УТВЕРЖДЕН
41327606.425000.463-01 34 01-ЛУ

Руководство оператора
«Speech - обработчик речевой информации»
(подсистема проекта SOVA)

41327606.425000.463-01 34 01

Листов 13

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

2020
www.sova.ai

АННОТАЦИЯ

Настоящее руководство предназначено для ознакомления оператора с функциональными и возможностями программы Speech. Программа предназначена для оператора, в должностные обязанности которого входит настройка и обучение основанных на принципах программ, использующих элементы искусственного интеллекта.

Руководство включает в себя описание технологии работы с программой, сведения о функциях программы, а также сведения о сообщениях, формирующихся при работе с программой.

В данном руководстве описаны особенности запуска, основные принципы управления программой, графический интерфейс пользователя, API, способы взаимодействия с другими программами.

Все важные моменты сопровождаются иллюстрациями и поясняющими примерами, что позволяет наглядно представить результат выполнения действий и сравнить изображение, выведенное на экран, с рисунками в руководстве. В настоящем руководстве в качестве иллюстраций используются экранные формы (окна), отображаемые на рабочем столе.

Настоящее руководство распространяется исключительно на программу и не заменяет учебную, справочную литературу, руководства от производителя ОС и прочие источники информации, освещающие работу с графическим пользовательским интерфейсом операционной системы.

СОДЕРЖАНИЕ

1 НАЗНАЧЕНИЕ ПРОГРАММЫ

1.1 Функциональное назначение программы

Подсистема Speech, содержит в себе модули распознавания (ASR – automatic speech recognition) и синтеза (TTS – text to speech) речи, позволяет выполнять полный спектр задач, связанных с обработкой и генерацией человеческой речи: распознавание речи в реальном времени с последующей обработкой полученного текста диалоговым процессором чат-бота, транскрибирование записей конференций, обработка типовых телефонных звонков в колл-центр, синтез ответных сообщений голосами выбранных дикторов, а также модификация сгенерированных записей путём контроля высоты тона и скорости.

1.2 Эксплуатационное назначение программы

Программа может встраиваться в сторонние приложения через интерфейс API, а также использоваться в качестве модуля, установленного в систему локально или доступного удаленно.

1.3 Состав функций программы

В состав Подсистемы Speech входят два модуля:

1 Модуль распознавания речи SOVA ASR имеет единственную функцию – перевод аудио записей в текст.

2 Модуль синтеза речи SOVA TTS имеет единственную функцию – перевод текста в аудио.

2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Для выполнения программы необходимо соблюдение условий минимального состава аппаратных и программных средств, а также подключенные к компьютеру устройства ввода (клавиатура, мышь), вывода изображения (монитор) и звука (динамики).

2.1 Минимальный состав аппаратных средств

Минимальные требования к оборудованию (ЭВМ):

- тактовая частота процессора – от 2 ГГц,
- количество ядер процессора – от 8 штук,
- от 16 Гб оперативной памяти,
- размер видео памяти – от 8 Гб,
- размер дисковый накопитель – от 50 Гб

2.2 Рекомендуемый состав аппаратных средств

Для устойчивой работы программы рекомендуется использовать ЭВМ с параметрами не ниже:

- Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz, x86_64, 8 cores
- 32 GB RAM
- NVIDIA Tesla T4

2.3 Минимальный состав программных средств

- Python (dev версии) версии не ниже 3.6
- Docker версии не ниже 19.03
- docker-compose версии не ниже 1.25
- CUDA версии не ниже 10.0, cuDNN, установленные драйверы NVIDIA.

2.4 Требования к персоналу (пользователю)

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц - системный программист и конечный пользователь программы - оператор.

Системный программист должен иметь техническое образование. В перечень задач, выполняемых системным программистом, должны входить:

1) задача поддержания работоспособности технических средств;

2) задачи установки (инсталляции) и поддержания работоспособности системных программных средств - операционной системы;

3) задача установки (инсталляции) программы. Ubuntu

3 ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1 Загрузка и запуск программы

Программные модули запускаются внутри докер контейнеров, для их запуска необходимо перейти в командной строке в директорию с развёрнутым модулем (SOVA ASR или SOVA TTS) и запустить команду **docker-compose up -d**. Далее можно отправлять запросы к API сервисов, либо открыть в браузере веб интерфейс запущенных приложений на заданных портах.

3.2 Выполнение функции программы

Пример запроса к API модуля SOVA ASR:

```
"request": {
        "auth": {
                "type": "noauth"
        },
        "method": "POST",
        "body": {
                "mode": "formdata",
                "formdata": [
                        {
                                "key": "audio_blob",
                                "type": "file",
                                "src": ""
                        }
                ]
        },
        "url": {
                "raw": "https://asr.ashmanov.org/asr/",
                "protocol": "https",
                "host": [
                        "asr",
```

```
                "ashmanov",
                "org"
            ],
            "path": [
                "asr",
                ""
            ]
        }
}
```

Формат ответа:

```
"response": {
    "r": [{
        "response_audio_url": "/media/0b9001cd-d4ac-4402-90f3-75cb31e5b1d0.wav",
        "response_code": 0,
        "response": [{
            "text": "мама мыла раму",
            "time": 0.478
        }]
    }]
}
```

В теле запроса можно менять передаваемый аудио блоб. Пример реализации средствами curl

Пример запроса к SOVA ASR с локальной машины:

```
$ curl --location --request POST 'http://localhost:8888/asr/' \
    --form 'audio_blob=@"/home/ubuntu/test.wav"'
```

Пример ответа:

```
{
    "r": [{
        "response_audio_url": "/media/0b9001cd-d4ac-4402-90f3-75cb31e5b1d0.wav",
        "response_code": 0,
        "response": [{
            "text": "привет",
            "time": 0.478
        }]
    }]
}
```

Пример запроса к API модуля SOVA TTS:

```
"request": {
    "method": "POST",
    "body": {
        "mode": "formdata",
        "formdata": [
            {
                "key": "model_type",
                "value": "Ruslan",
                "type": "text"
            },
            {
                "key": "text",
                "value": "Привет, мир.",
                "type": "text"
            },
            {
                "key": "options",
                "value": "",
                "type": "text"
            }
        ]
    },
    "url": {
        "raw": "https://tts-test.ashmanov.org/tts/",
        "protocol": "https",
        "host": [
            "tts-test",
            "ashmanov",
            "org"
        ],
        "path": [
            "tts",
            ""
        ]
    }
```

```
}
```

Формат ответа:

```
{
    "response":[
    {
        "name": "Ruslan",
        "time": 0.778
    }
    ],
    "response_audio": …,
    "response_audio_url":"/media/data/waves/ruslan_38be3647b0931e_2020-08-11_19-25.wav",
    "response_code": 0
}
```

Здесь можно изменять входную текстовую строку и указать текст, который требуется озвучить.

Пример запроса на Python с сохранение получаемого аудио блоба как файла:

```
import json
import requests
from base64 import b64decode



def tts(text):
    url = "https://tts-test.ashmanov.org/tts/"
    payload = {"model_type": "Tacotron2", "text": text}

    response = requests.request("POST", url, headers={}, data=payload)

    if response:
        return b64decode(json.loads(response.text)["response_audio"].encode("utf-8"))
```

```
        else:
            return False



def main():
    text = "д+обрый д+ень"
    audio = tts(text)
    with open("result.wav", "wb") as f:
        f.write(audio)



if __name__=="__main__":
    main()
```

3.3 Завершение работы программы

Программные модули запускаются внутри докер контейнеров и останавливаются средствами Docker: для того, чтобы остановить работу контейнера, необходимо узнать его ID, для этого можно использовать команду **docker ps**. Узнав ID контейнера – можно остановить его работу командой **docker kill** *id_контейнера*.

4 СООБЩЕНИЯ ОПЕРАТОРУ

Программные модули запускаются внутри докер контейнеров и подробное логгирование остаётся внутри контейнеров, просмотреть логи можно узнав ID контейнера использовав команду **docker ps**, после чего выполнить команду **docker logs** *id_контейнера*. В выводе предусмотрены сообщения для отладки приложения в случае возникновения ошибок, которые может разрешить системный программист.

5 ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

- ASR (Automated Speech Recognition) — автоматическое распознавание речи;
- Speech язык разметки для приложений синтеза речи;
- TTS (Text To Speech) - технология, которая позволяет преобразовывать текстовые файлы в аудиофайлы при помощи анализа грамматических структур текста;
- KenLM - это быстрый набор инструментов для моделирования языка с низким объемом памяти, который масштабируется до триллионов слов;

- CPU - Центральный процессор — электронный блок либо интегральная схема, исполняющая машинные инструкции (код программ), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера;
- Нейронная сеть также искусственная нейронная сеть, ИНС — математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей - сетей нервных клеток живого организма;
- Лексико-смысловой анализатор - система автоматического понимания текстов, заключающаяся в выделении семантических отношений, формировании семантического представления текстов;
- API - (программный интерфейс приложения, интерфейс прикладного программирования) (англ. application programming interface) — описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой,
- CUID- уникальный идентификатор чата, используется для идентификации чата, к которому относится очередной запрос СУБД
- Руководство системного программиста - перечень данных включающий в себя: общие сведения о программе, структуру программы, настройку программы, проверку программы, дополнительные возможности и сообщения системному программисту выдаваемых в ходе выполнения настройки, проверки программы, а также в ходе выполнения программы, описание их содержания и действий, которые необходимо предпринять по этим сообщениям;
- Руководство оператора - перечень данных, включающий в себя: назначение программы, условия выполнения программы, выполнение программы, сообщения оператору;
- Программа и методики испытаний - организационно-методический документ, обязательный к выполнению, включающий метод испытаний, средства и условия испытаний, алгоритмы выполнения операций по определению одной или нескольких взаимосвязанных характеристик свойств объекта, формы представления данных и оценивания точности, достоверности результатов;
- Ведомость эксплуатационных документов - учетный документ, составленный в виде списка эксплуатационных документов;
- ГОСТ- Межгосударственный стандарт — региональный стандарт, принятый Межгосударственным советом по стандартизации, метрологии и сертификации Содружества независимых государств;
- Техническое задание - перечень требований, условий, целей, задач, поставленных заказчиком в письменном виде, документально оформленных и выданных исполнителю работ проектно-исследовательского характер.

Оформление настоящего документа произведено в соответствии с ГОСТ 19.505 79.

6 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Shen, Jonathan, Ruoming Pang, Ron J. Weiss, Michael Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis and Yonghui Wu. “Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions.” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018): 4779-4783.
2. Prenger, Ryan, Rafael Valle and Bryan Catanzaro. “Waveglow: A Flow-based Generative Network for Speech Synthesis.” *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019): 3617-3621.
3. Wang, Yuxuan, Daisy Stanton, Yu Zhang, R. J. Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia and Rif A. Saurous. “Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis.” *ArXiv* abs/1803.09017 (2018).
4. Zhu, Xiaolian, Y. Zhang, Shan Yang, Liumeng Xue and L. Xie. “Pre-Alignment Guided Attention for Improving Training Efficiency and Model Stability in End-to-End Speech Synthesis.” *IEEE Access* 7 (2019): 65955-65964.
5. Liu, Peng, Xixin Wu, Shiyin Kang, G. Li, Dan Su and Dong Yu. “Maximizing Mutual Information for Tacotron.” *ArXiv* abs/1909.01145 (2019).
6. Vineel Pratap, Awni Hannun, Qiantong Xu, Jeff Cai, Jacob Kahn, Gabriel Synnaeve, Vitaliy Liptchinsky, Ronan Collobert, “wav2letter++: The Fastest Open-source Speech Recognition System”, arXiv:1812.07625, 2018.
7. Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve, “Wav2letter: an end-to-end convnet-based speech recognition system,” *CoRR*, vol. abs/1609.03193, 2016.
8. Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert, “Letter-based speech recognition with gated convnets,” *CoRR*, vol. abs/1712.09444, 2017.
9. Matteo Frigo and Steven G. Johnson, “The design and implementation of FFTW3,” Proceedings of the IEEE, vol. 93, no. 2, pp. 216–231, 2005, Special issue on “Program Generation, Optimization, and Platform Adaptation”.
10. Alex Graves, Santiago Ferna´ndez, Faustino Gomez, and Ju¨rgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 369–376.

Лист регистрации изменений									
Номера листов (страниц)					Всего листов (страниц) в докум.	№ документа	Входящий № сопроводит. докум. и дата	Подп.	Дата
Изм.	измененных	замененных	новых	аннулированных					